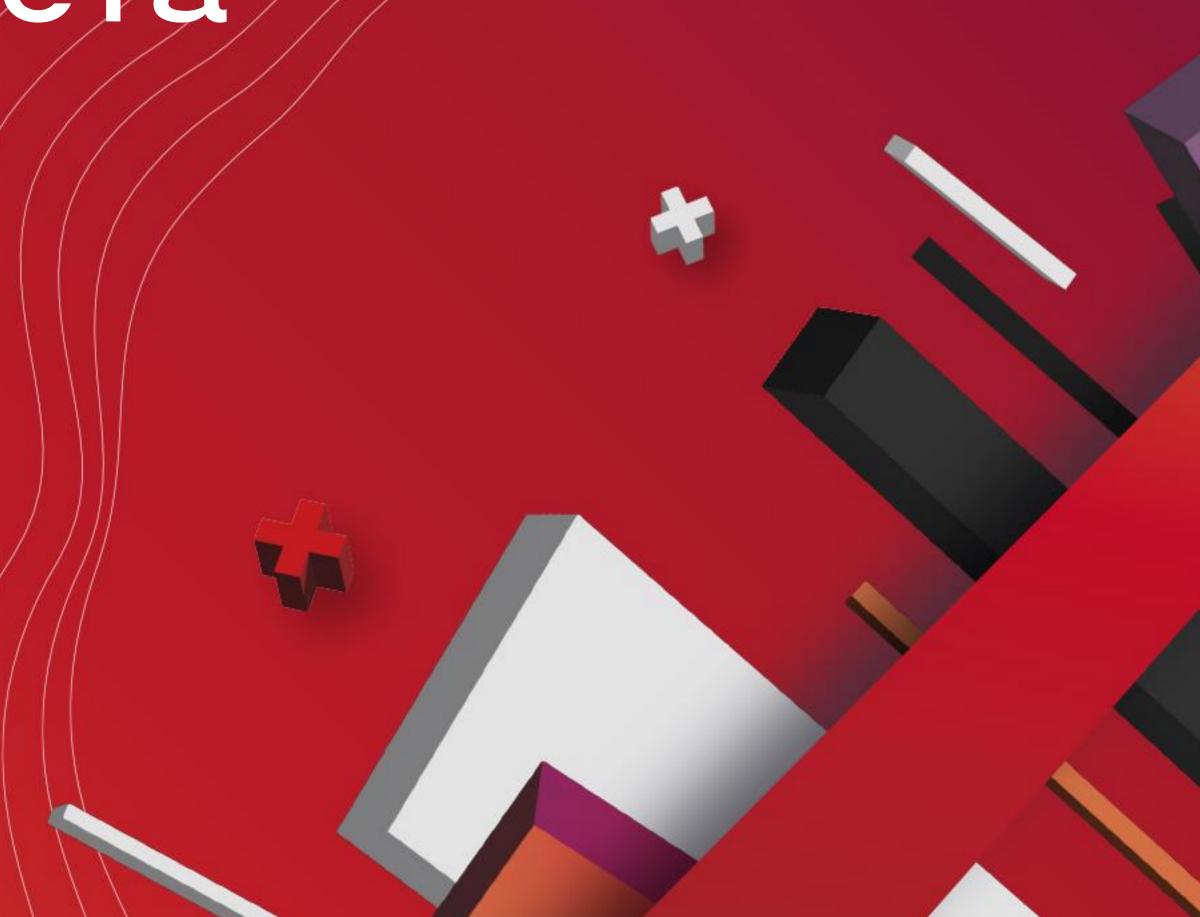
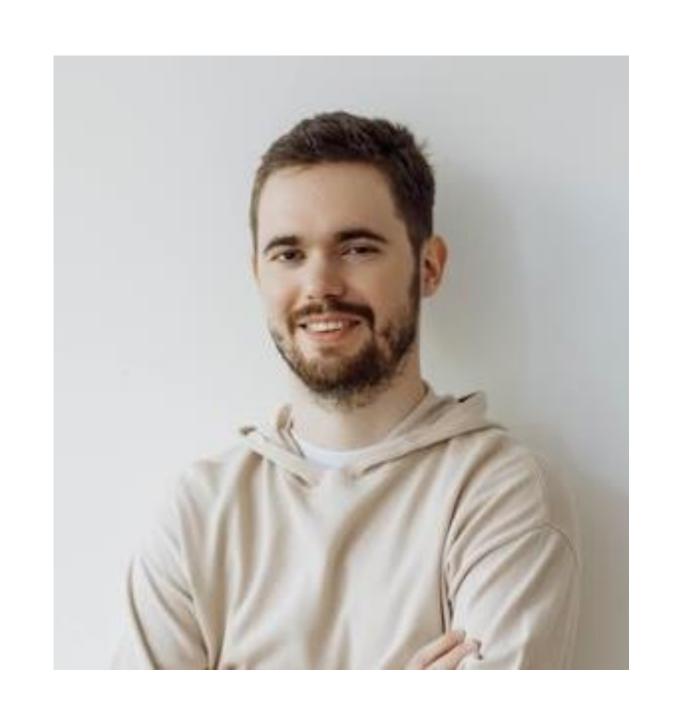


Кучумов Илья, руководитель разработки Товарного Поиска



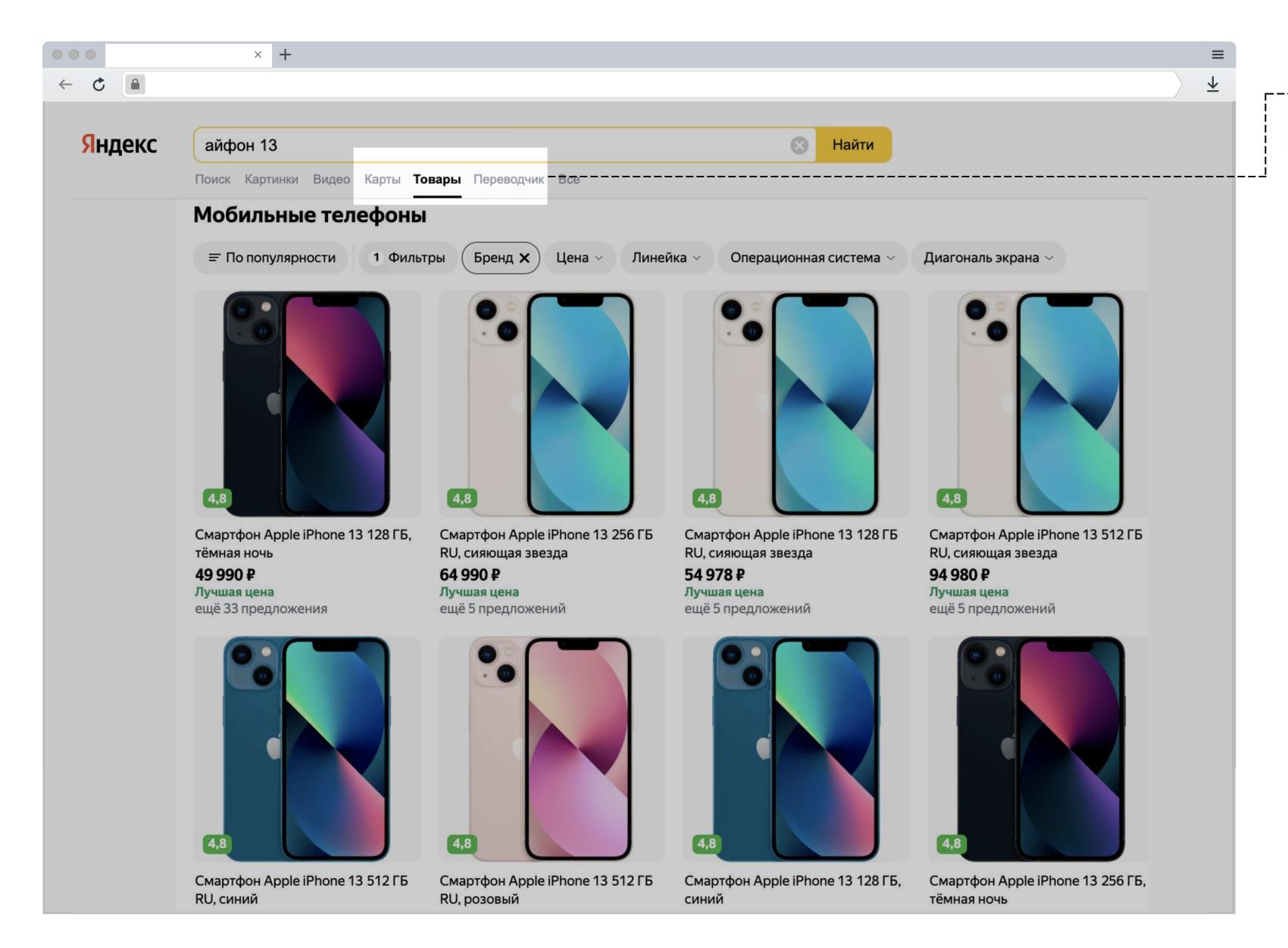


Обо мне



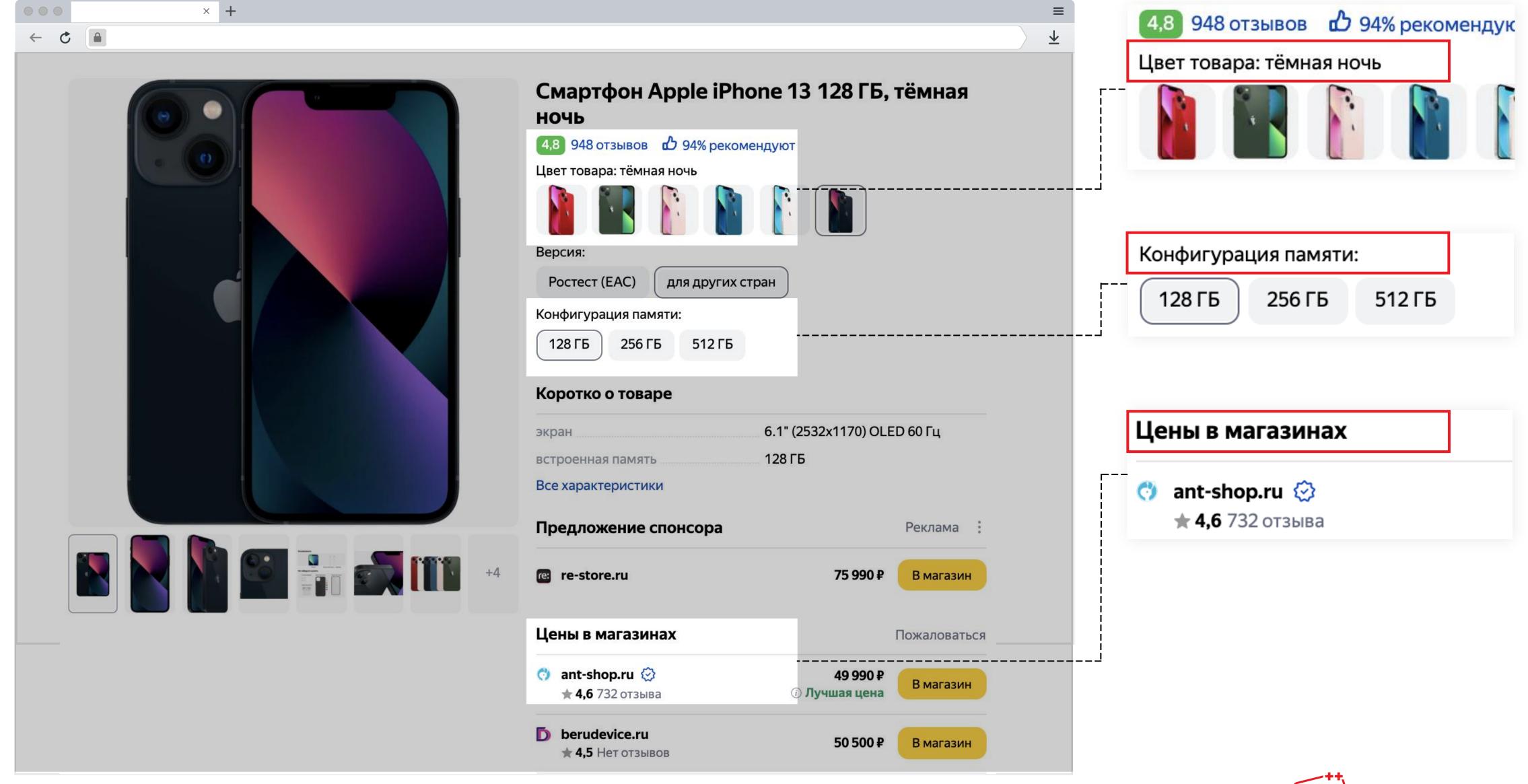
- Руководитель отдела разработки
 Товарного Поиска в Яндексе
- + В прошлом руководил качеством поиска в Лавке и разработкой Турбо-страниц













Про что доклад

- + Про построение базы цен Товарного Поиска
- + Мы проверили, все работает и на других срезах
- + Ключевая функциональность, высокие требования правильности
- + Цены быстро меняются, невозможно кэшировать



План доклада

- 01 Общая архитектура базы Товарного Поиска
- 02 Подробно про парсинг и скачивание
- 03 Другие удачные применения парсинга

Что такое построение базы цен

МЛРДТоваров

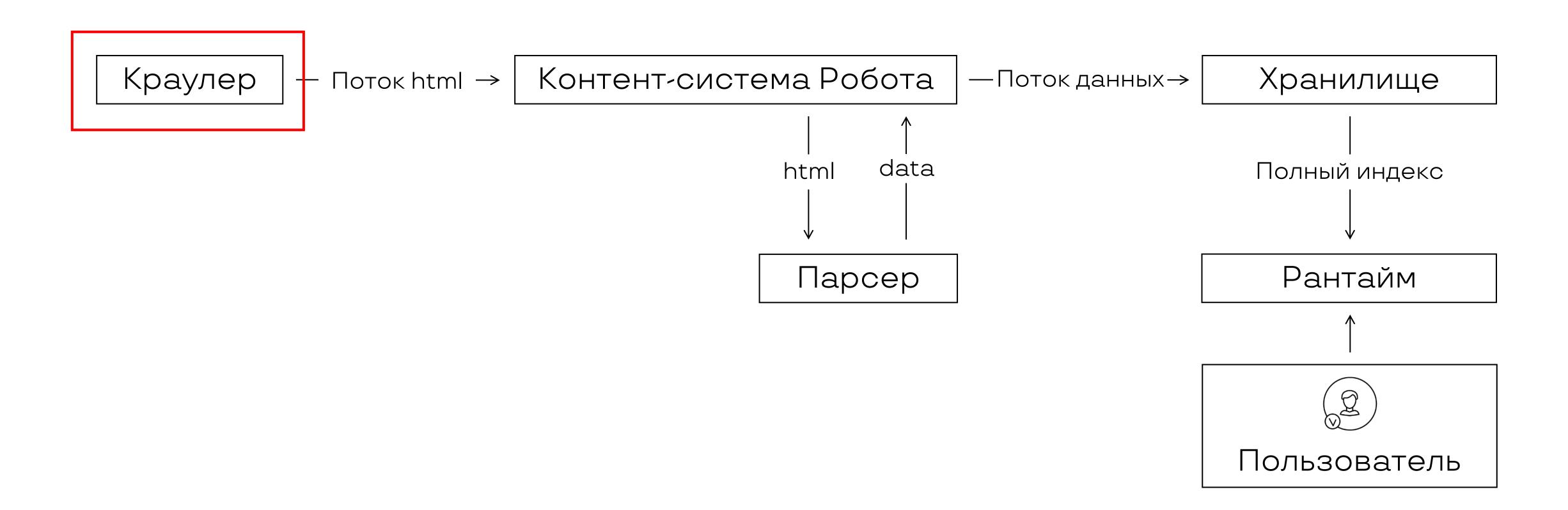
1 Б млн интернет-магазинов

1 о случайных товаров меняют цену каждые 24 часа

Ошибка цены — это плохо

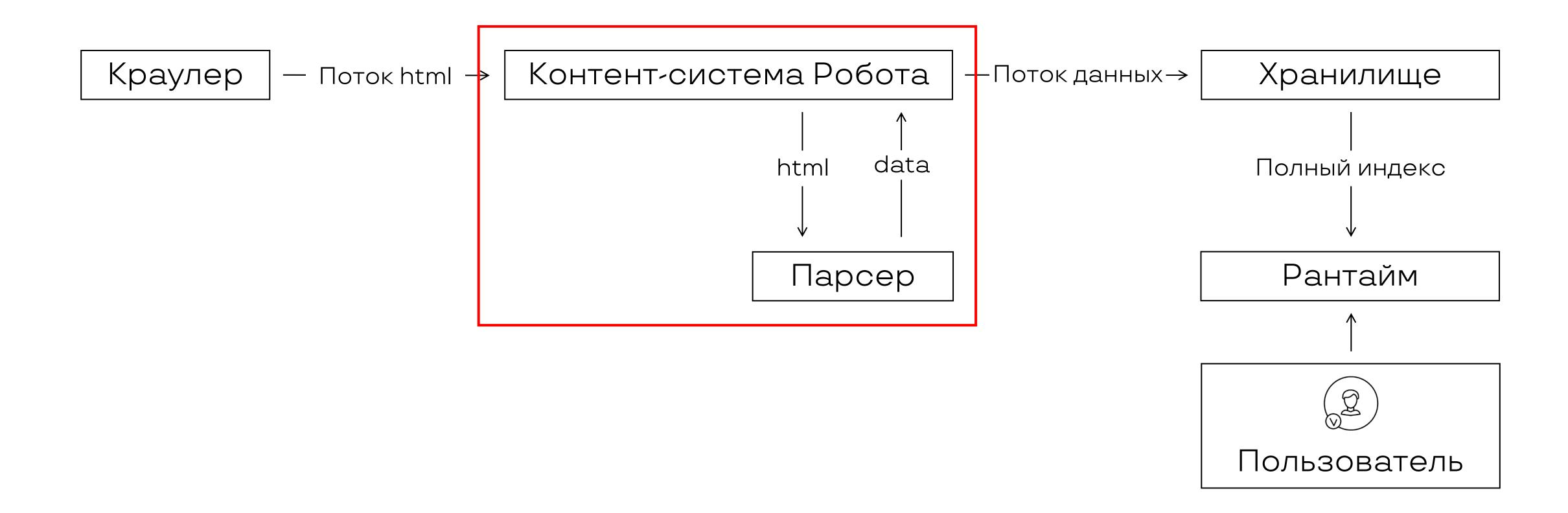


Общая архитектура



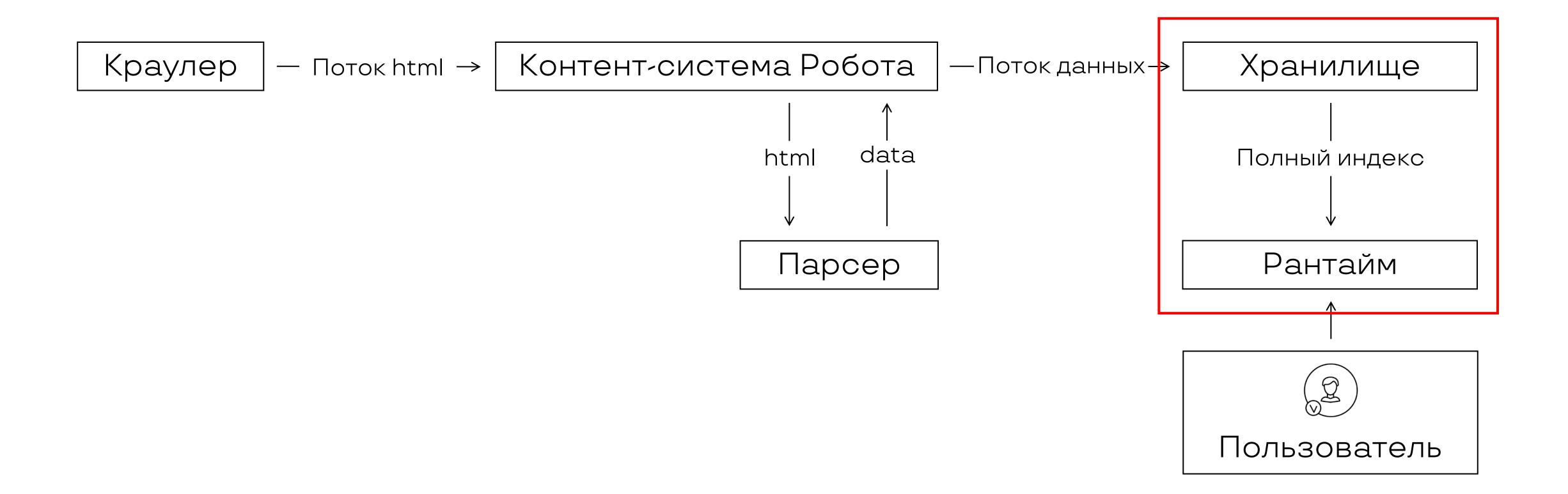


Общая архитектура



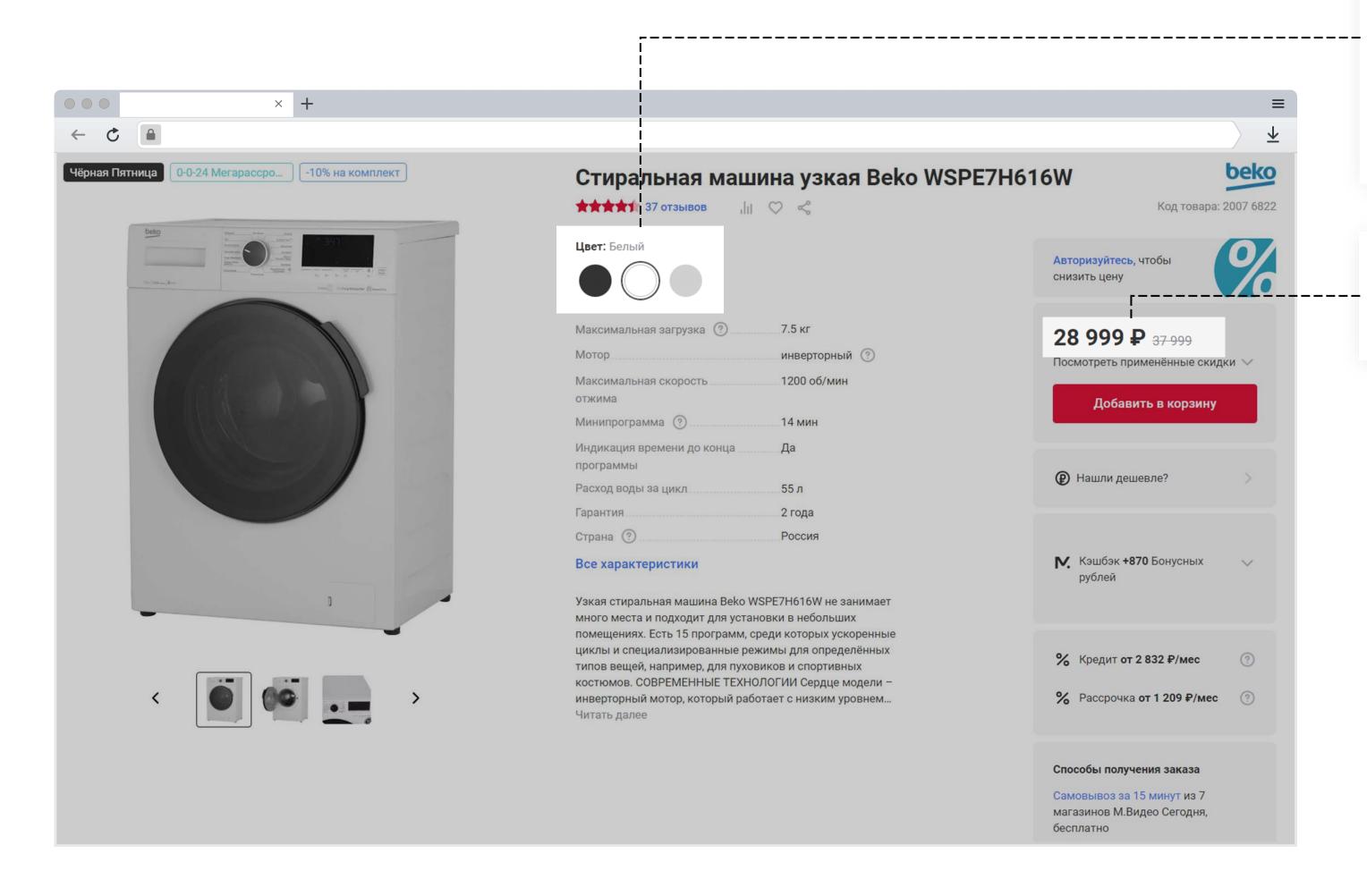


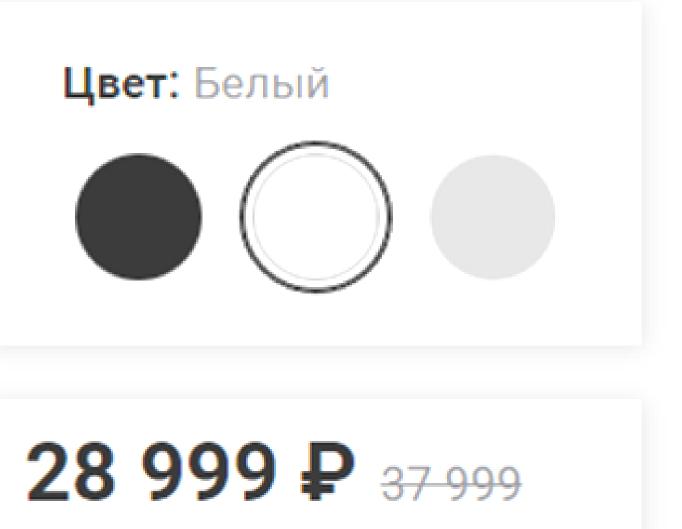
Общая архитектура





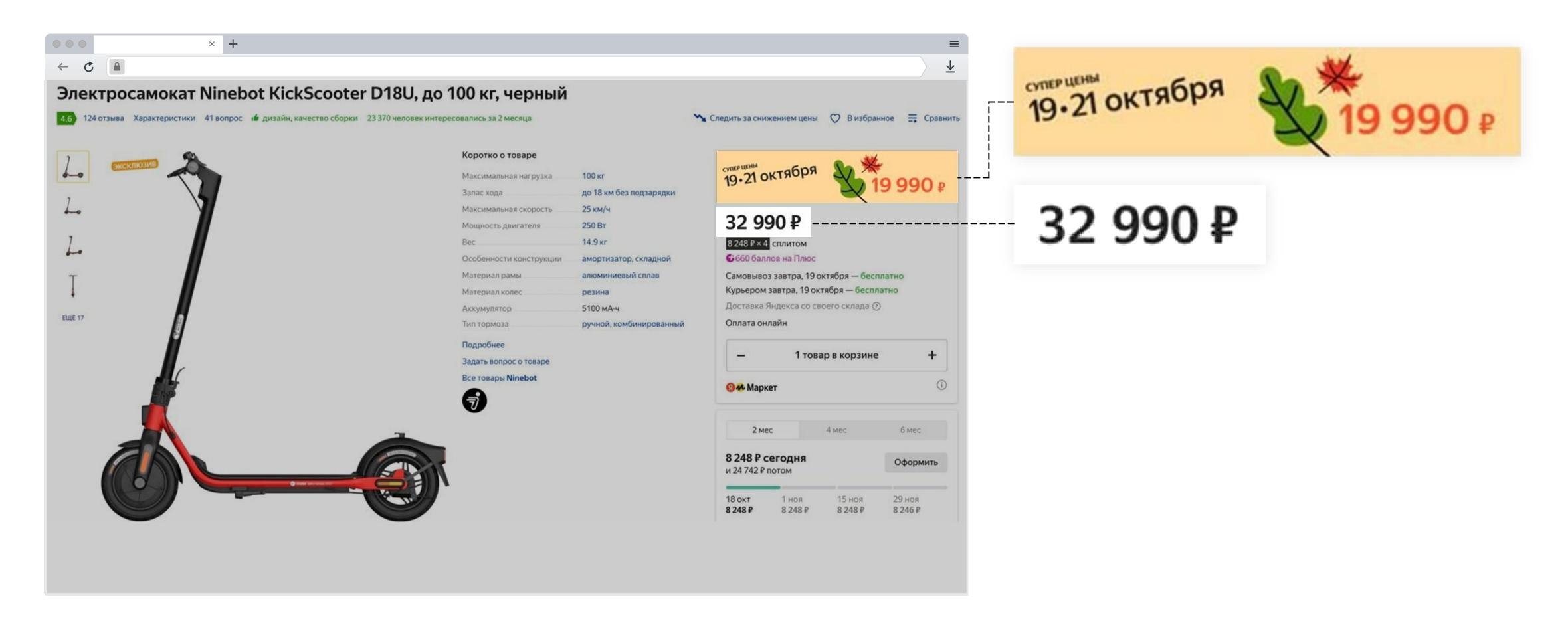
Что такое парсинг







Что такое парсинг





Что такое парсинг

```
{
    "title": "Стиральная машина",
    "price": 32999,
    "available": true,
    "main_image": "https:cdn.shop.ru/wash.jpg",
    "brand": "Самый лучший"
}
```



А можно ли просто?



Микроразметка

```
<meta itemprop="availability" content="https://schema.org/InStock" />
<meta itemprop="priceCurrency" content="RUB" />
<meta itemprop="price" content="82990" />
...
```

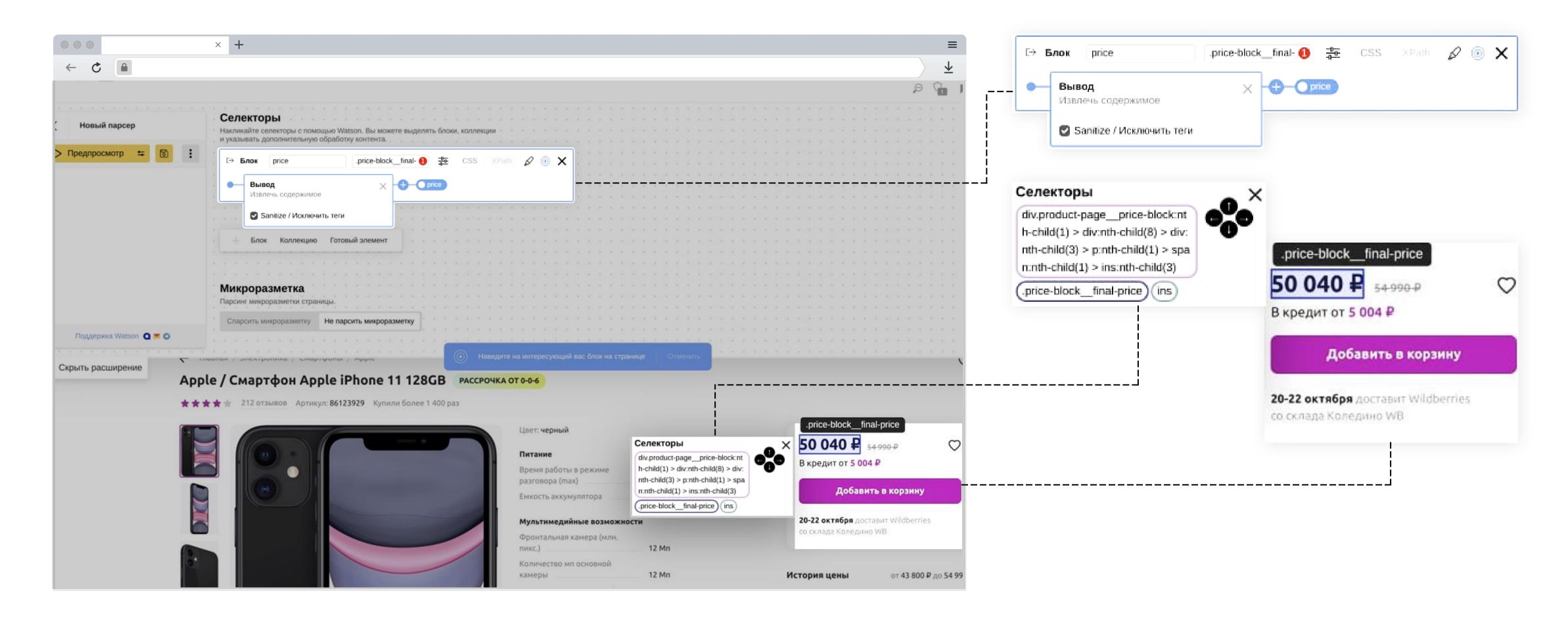
- + Быстро реализовать
- + Веб-мастера напрямую влияют
- + Покрытие 50% товаров
- + Невозможно влиять со стороны парсера



Парсеры на селекторах



Парсеры на селекторах





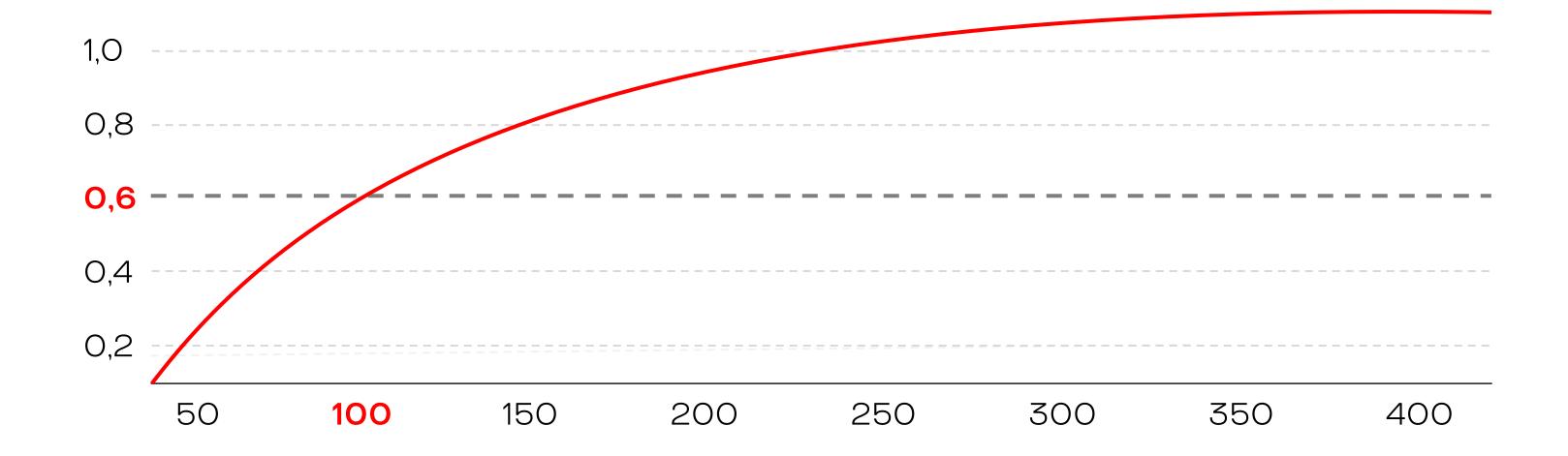
Попытка накликать парсеры

Apple **iPhone 13** Pro **Max 256**GB Dual SIM Silver купить...

best-magazin.com > Apple iPhone 13 Pro Max 256GB Dual SIM Silver (... ★4,9 Купить Айфон 13 Про Макс 256 ГБ Dual SIM в Бест-магазин. ... Процессор iPhone 13 Pro Max Dual SIM (256GB, Silver, 2 nano SIM) имеет 2 высокопроизводительных ядра (тактовая частота до 3,23 ГГц) плюс 4 ядра эффективности. Читать ещё



99 990 ₽





Попытка накликать парсеры

10 разработчиков

Б пицц

- + Кола и чай
- + Большая переговорка

Мы написали 100 **80 парсеров за встречу**

80 70 отправили в эксперимент (через месяц осталось 60)



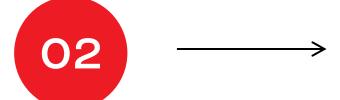
Парсеры на селекторах

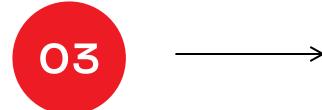
- + Возможность влиять на качество со стороны парсера
- + Быстро распарсить конкретный сайт
- + 1.5 млн. парсеров = около 130 рабочих человеко-лет накликивания
- + Нужна постоянная поддержка разработчиков



Парсеры краудом







04

Берем конкретный сайт Толокер создает селекторы Другие толокеры проверяют качество на семпле страниц

Если всё хорошо — добавляем в стейт парсера



Почему неидеальное решение?

+ Парсеры независимые

1.5 млн. = 1.1 млн. долларов О,75 долларов/штука

30% случайных парсеров сломаются за месяц

- Отоимость проверки 750 тыс. долларов (еженедельно)
- Расходы не делятся на другие срезы

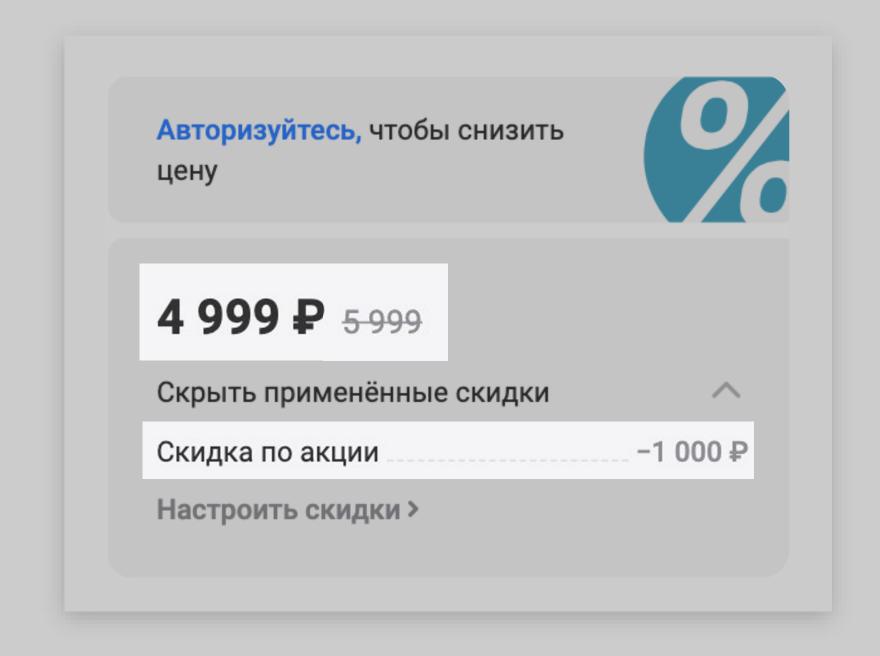


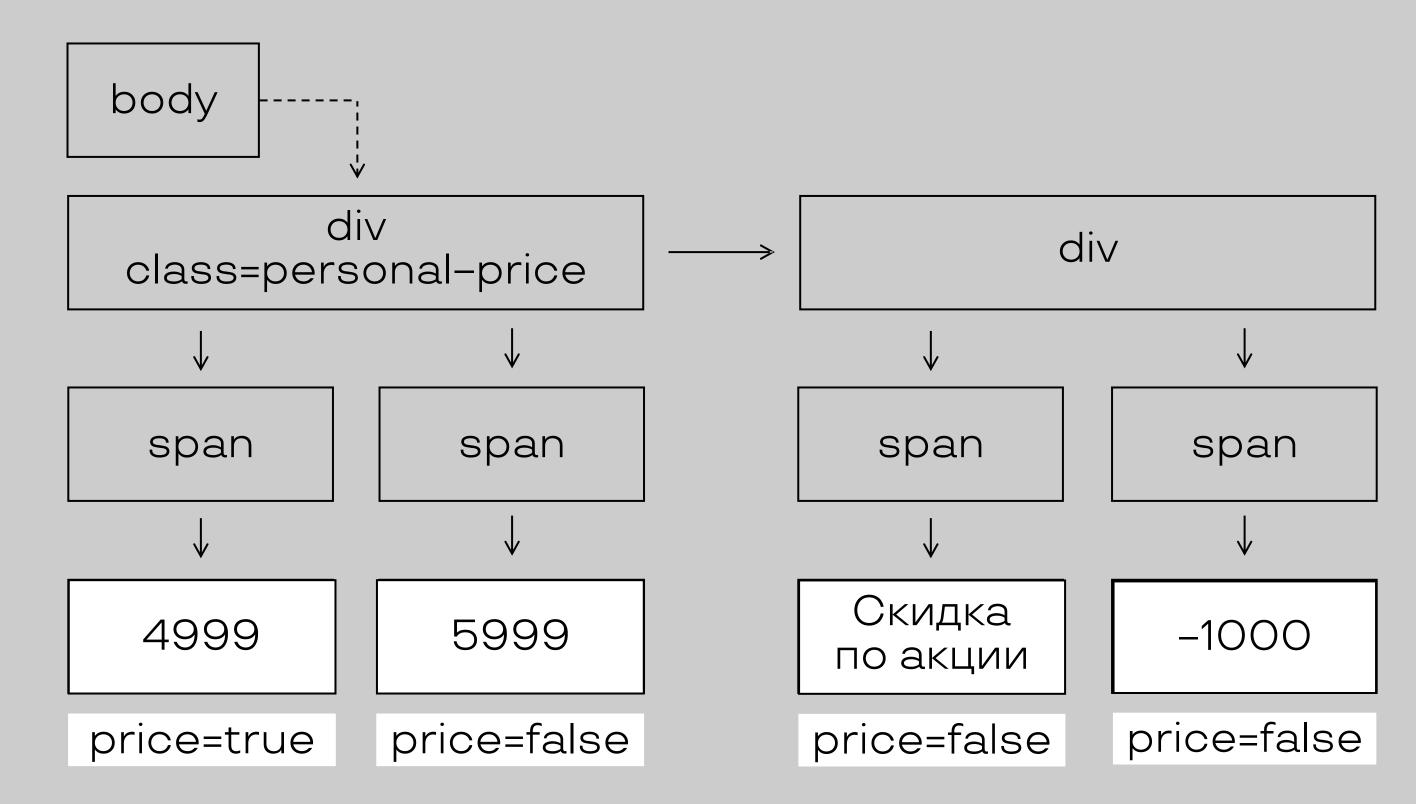
Саммари простых методов

- + Простых рабочих эвристик по контенту нет
- + Микроразметка среднее покрытие, но быстро запустить
- + Селекторы хорошее покрытие, но дорого поддерживать



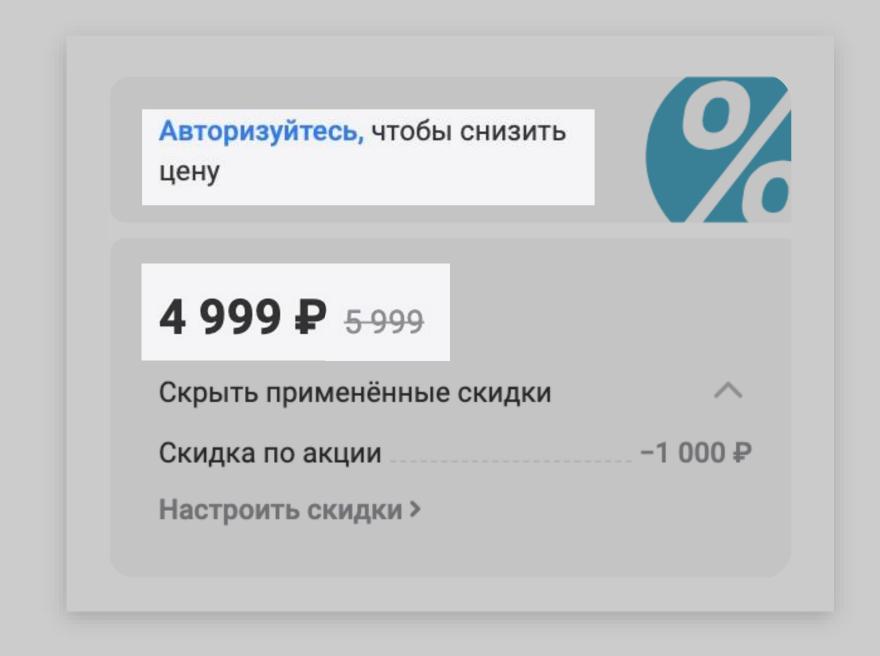
Классификация вершин DOM-дерева

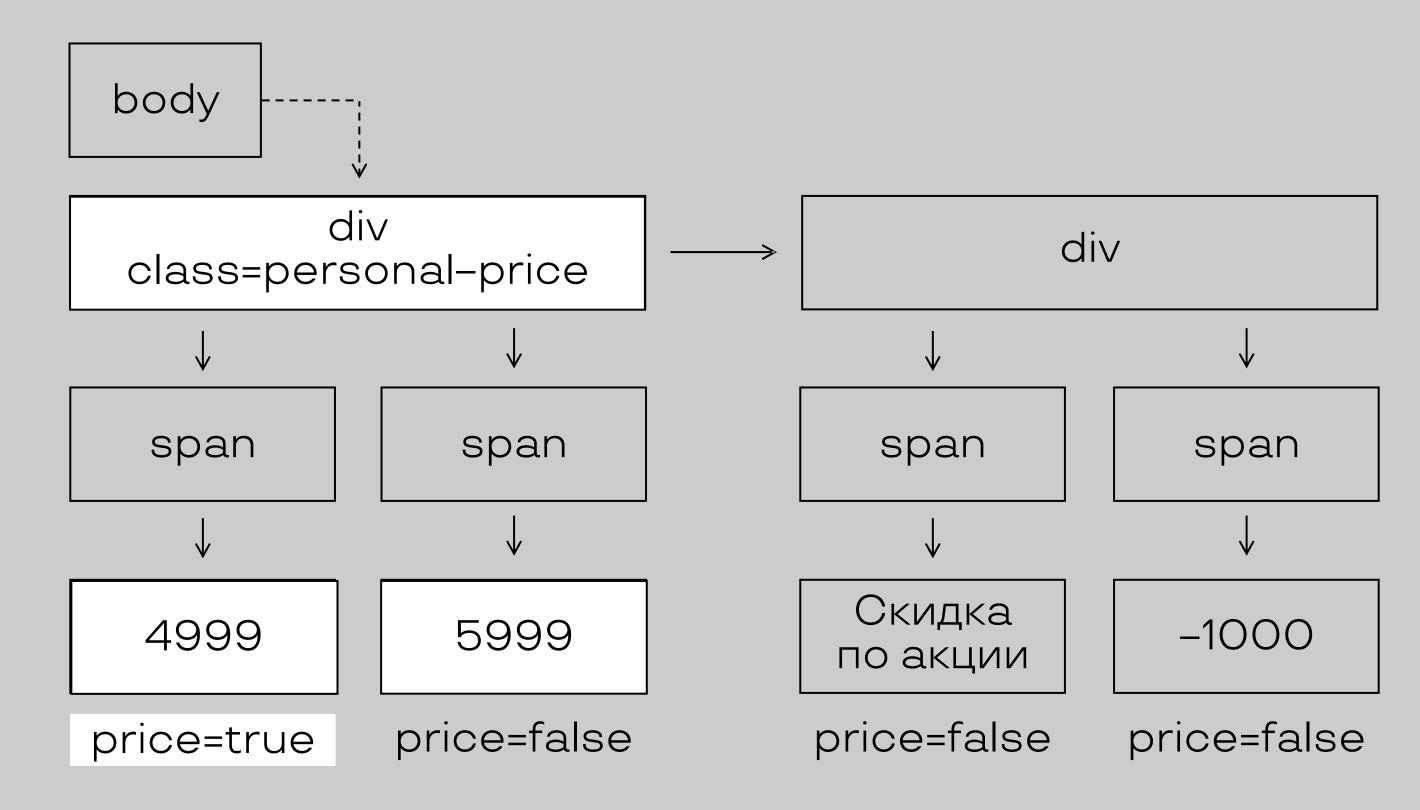






Текстовый фактор





Text = Авторизуйтесь, чтобы снизить цену | 4 999 || class=personal-price | 5999



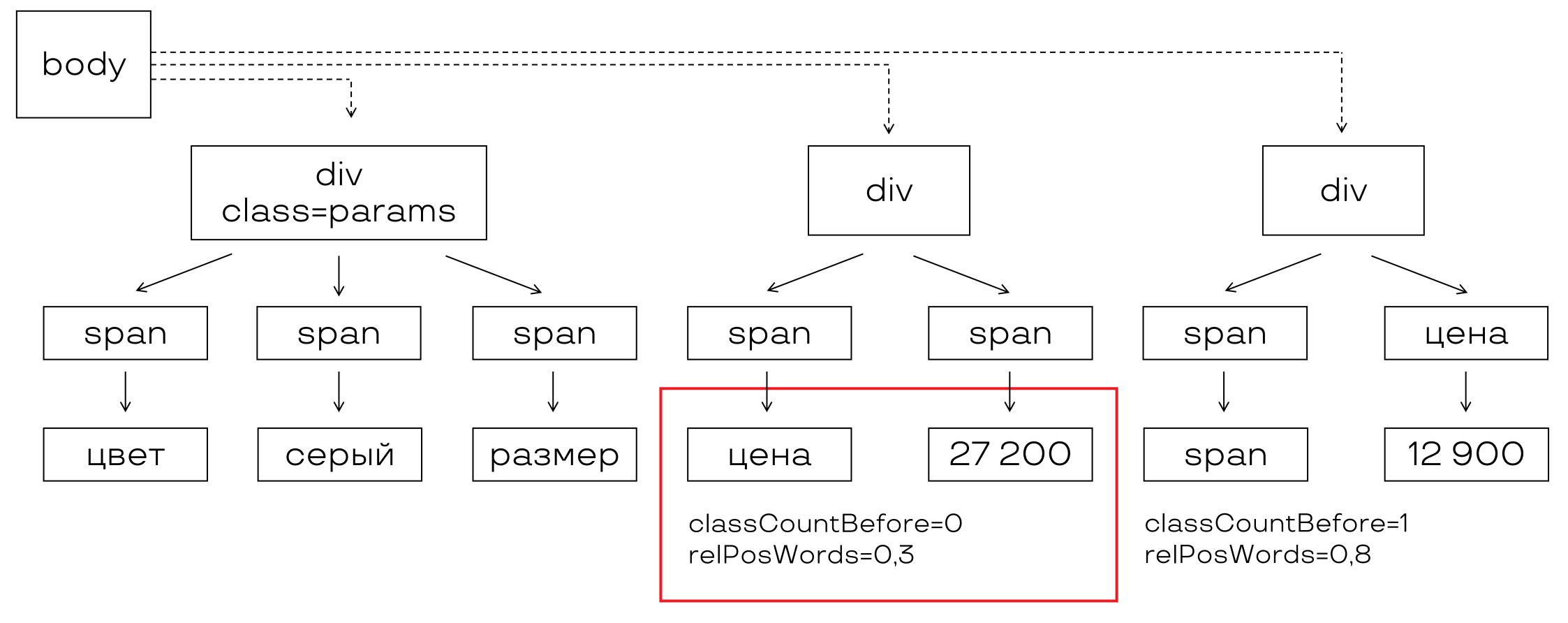


Контекст вершины в дереве

- + Тегродителя, дедушки, ...
- + Сколько параграфов (ссылок, картинок, ...) у родителя, дедушки?
- + А сколько еще таких же вершин в дереве?
- + Количество вершин у родителя, дедушки, ...

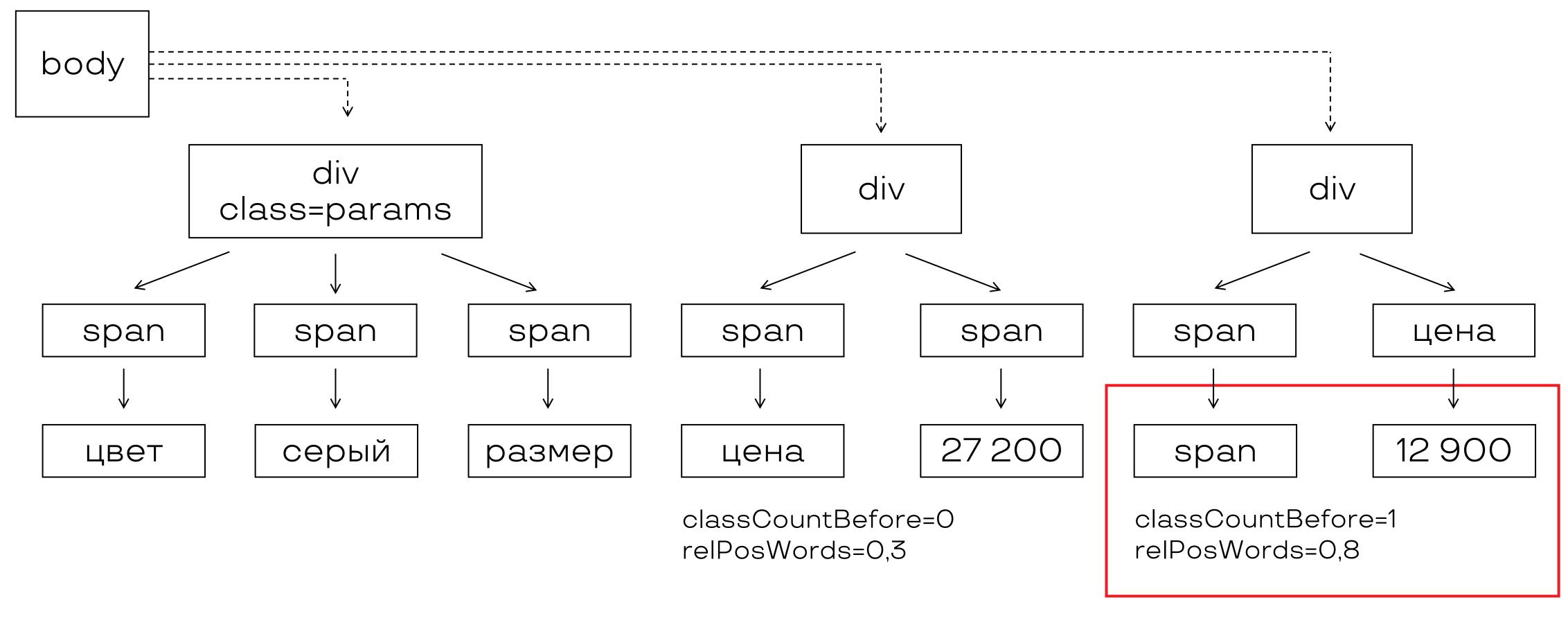


Позиция в дереве



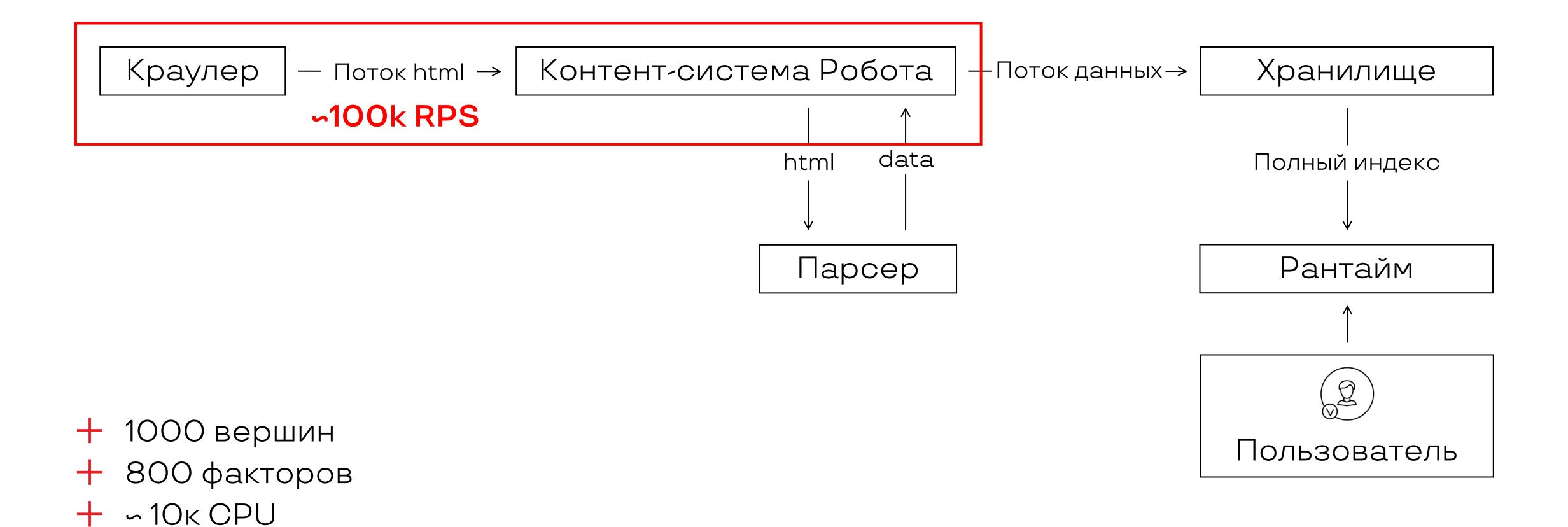


Позиция в дереве



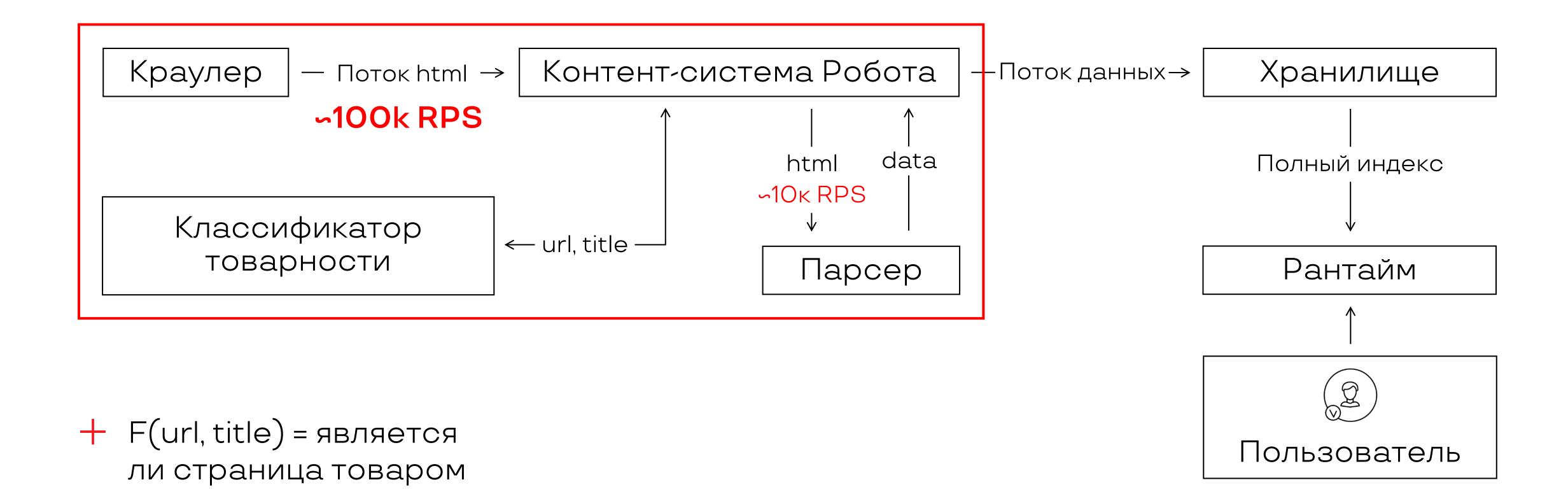


Необходимость оптимизаций





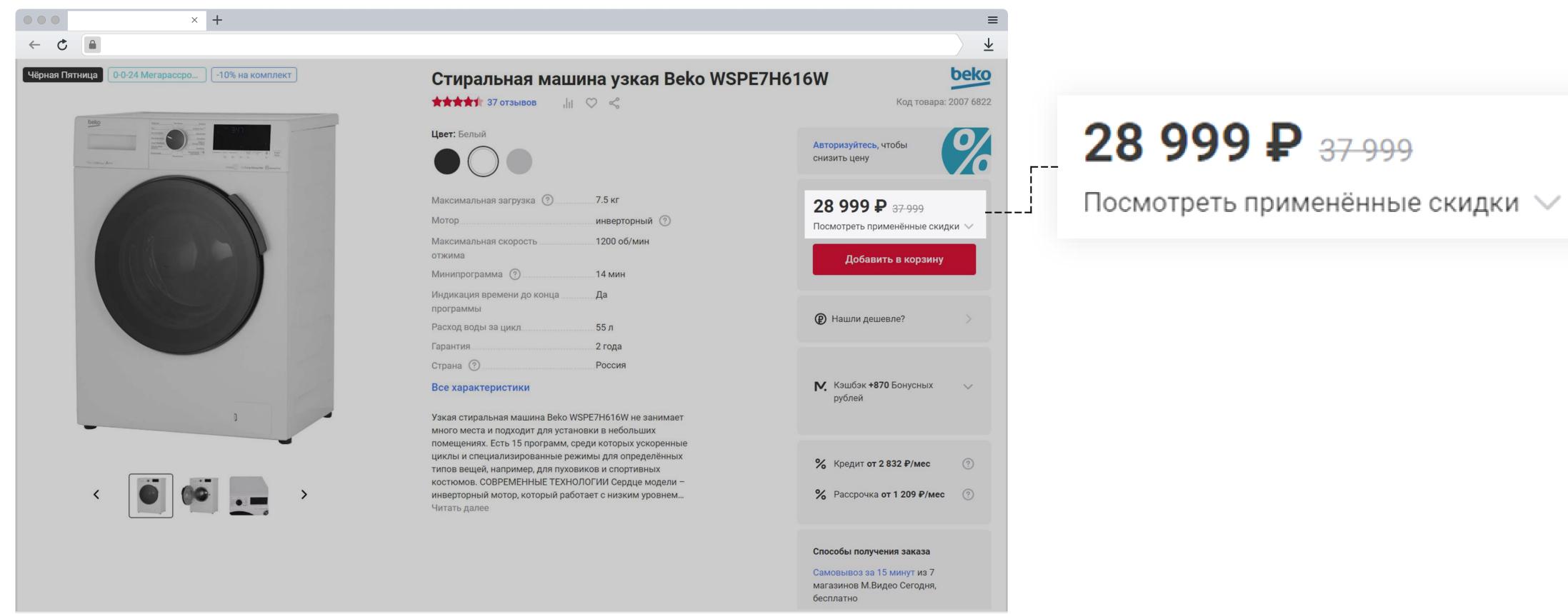
Сужение потока до товарных страниц





+ Легкая DSSM'ка

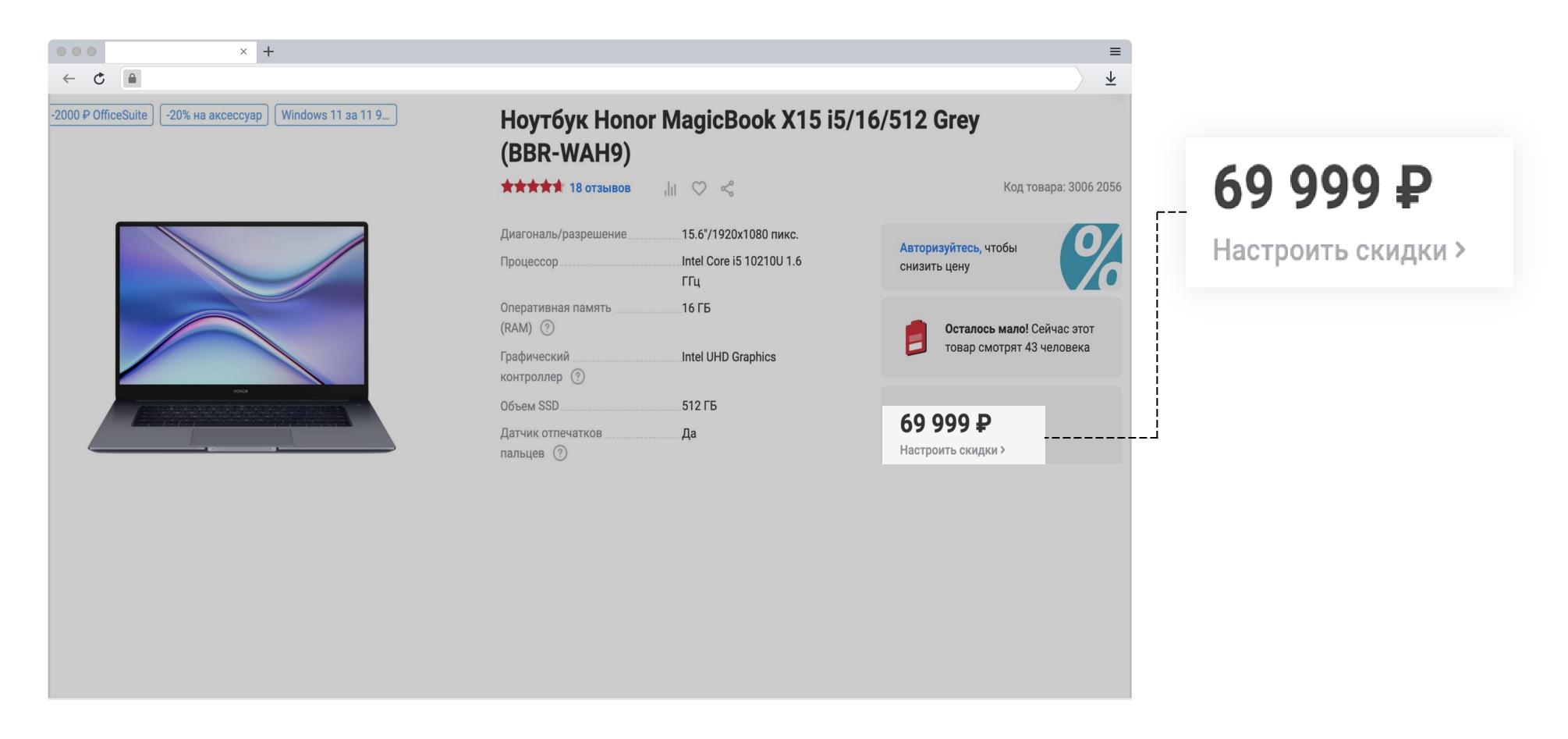
Похостовое кэширование







Похостовое кэширование



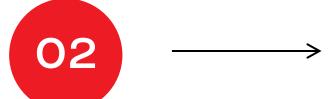


Давайте генерировать селекторы моделью



Алгоритм построения селекторов







04

Смотрим путь от вершины до корня Выписываем теги + класс от вершины до корня Проверяем, что все 99% вершин с таким селектором являются ценой

Сохраняем селектор для конкретного сайта



Как используем

- + Для каждого хоста (host, css_selector) → answer
- + Файлик с миллионом записей
- + Регулярно перестраиваем без участия крауда
- + Экономия 60% железа



Как режется поток

100к RPS

-90% за счёт классификатора

10k RPS

-50% за счёт микроразметки

5к RPS

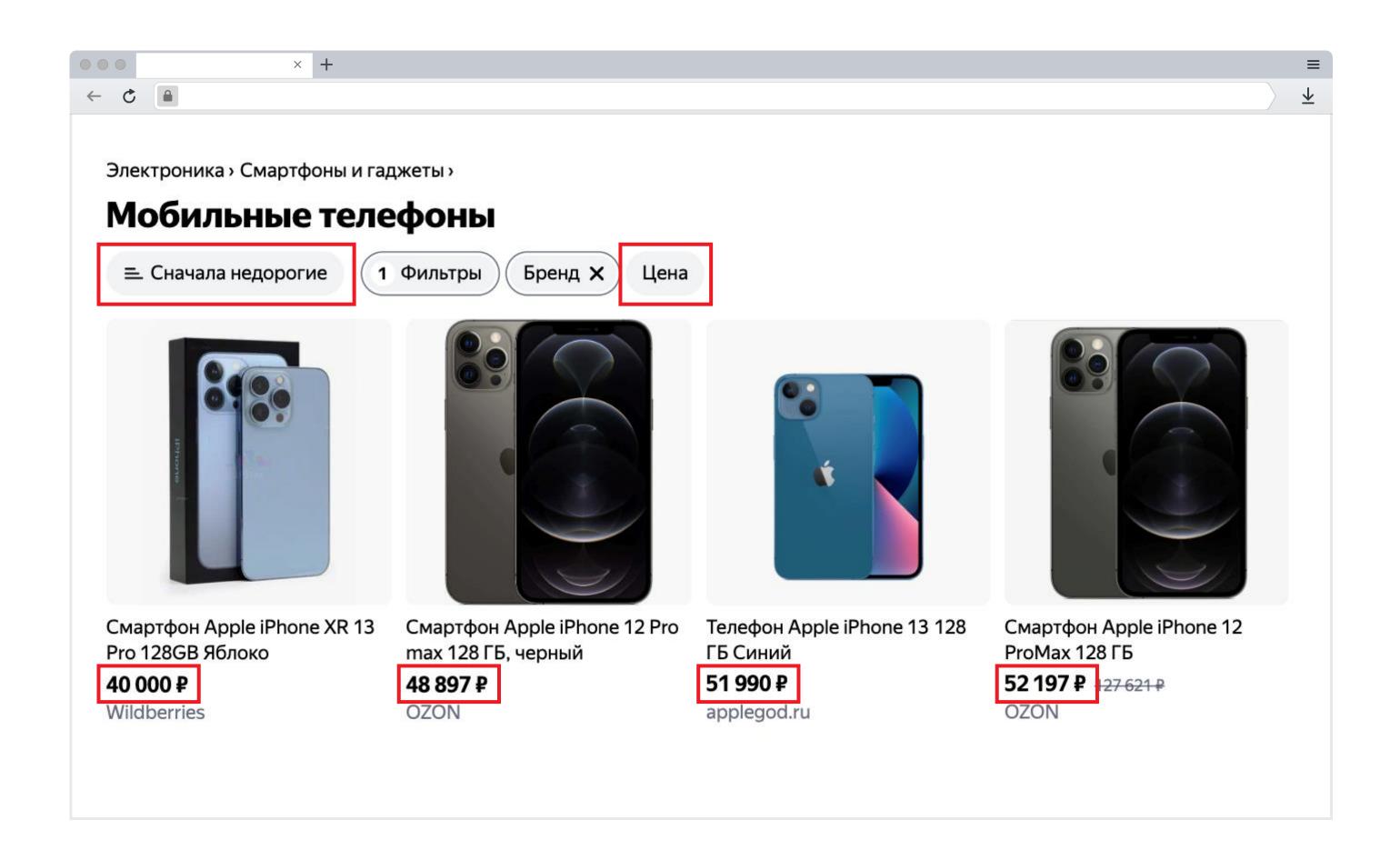
-60% за счёт кэширования

2k RPS

можно и распарсить



Почему актуальность цен важна





Давайте будем обходить базу раз в день (неделю?)



Поддержка актуальности

- + Round robin не работает, прокачка всей базы больше месяца
- + Похостовые лимиты серьёзная проблема
- + 10М за день это 116 RPS на конкретный сайт

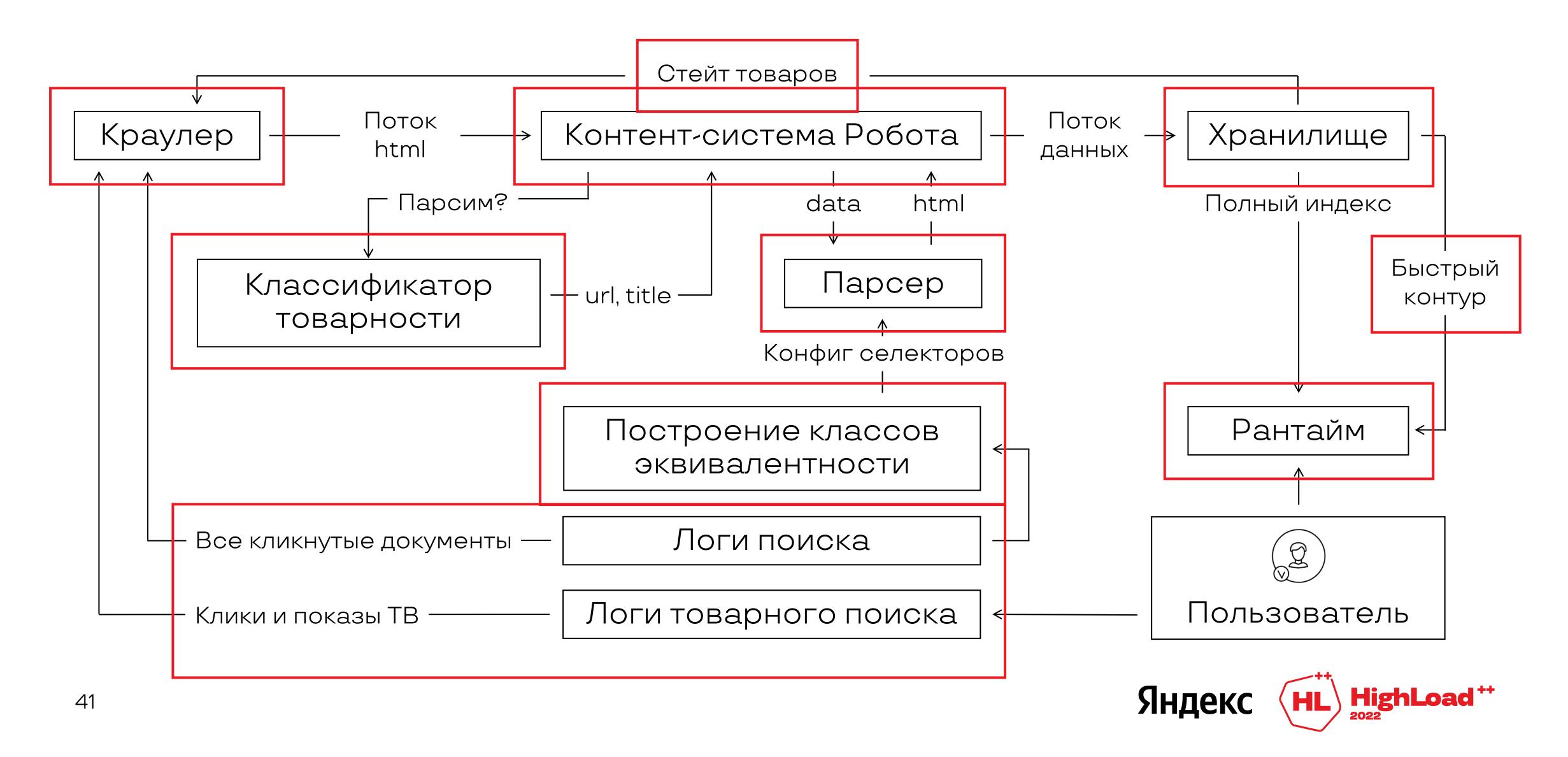


Эвристики скачивания

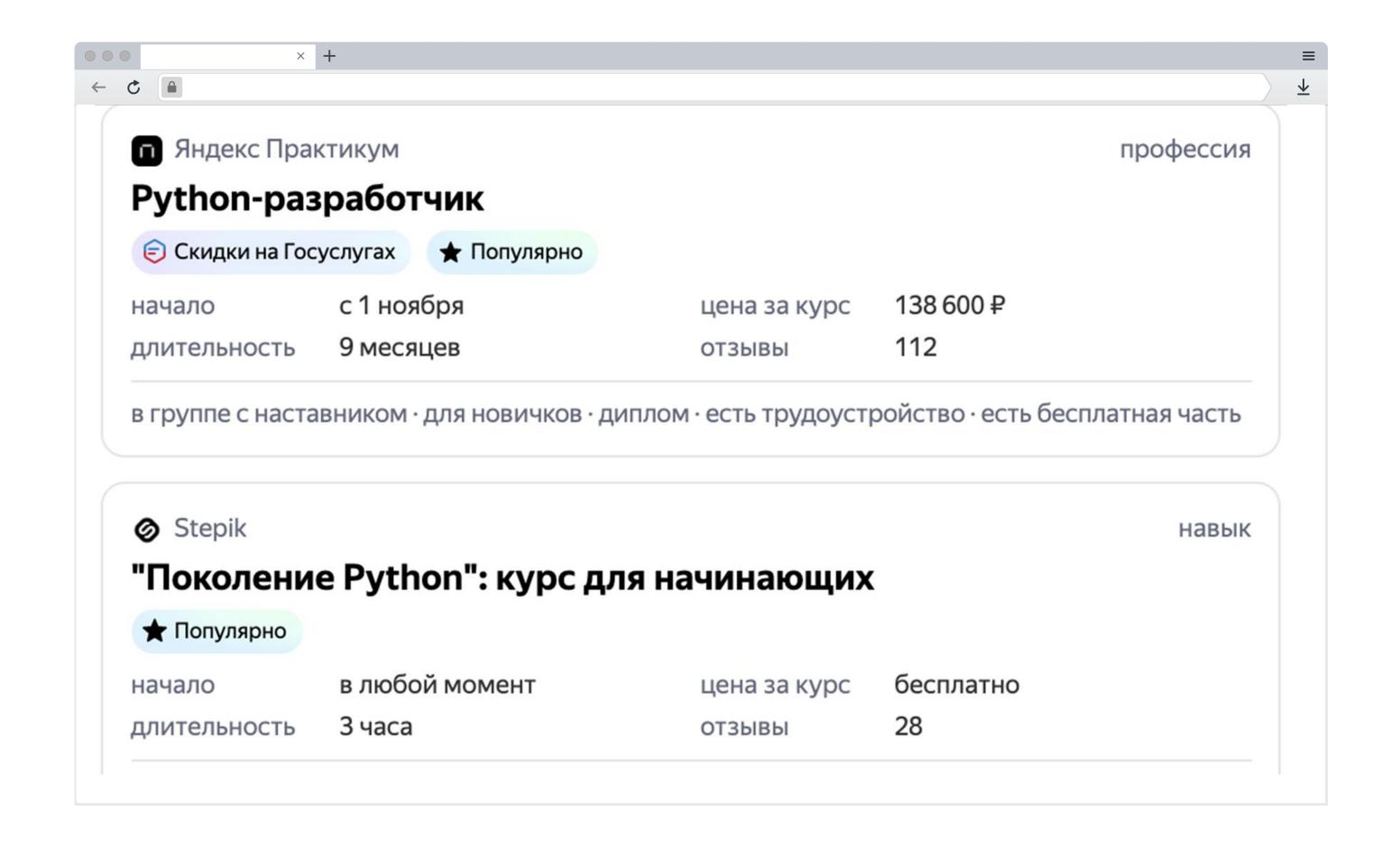
- + Показы в поиске обходим всегда покрываем популярные
- + По (title, category, offer_age) предсказываем вероятность изменения цены с последнего скачивания
- + Качаем самые вероятно изменившиеся
- + Подбираем фильтр по возрасту, если нужно



Финальная схема

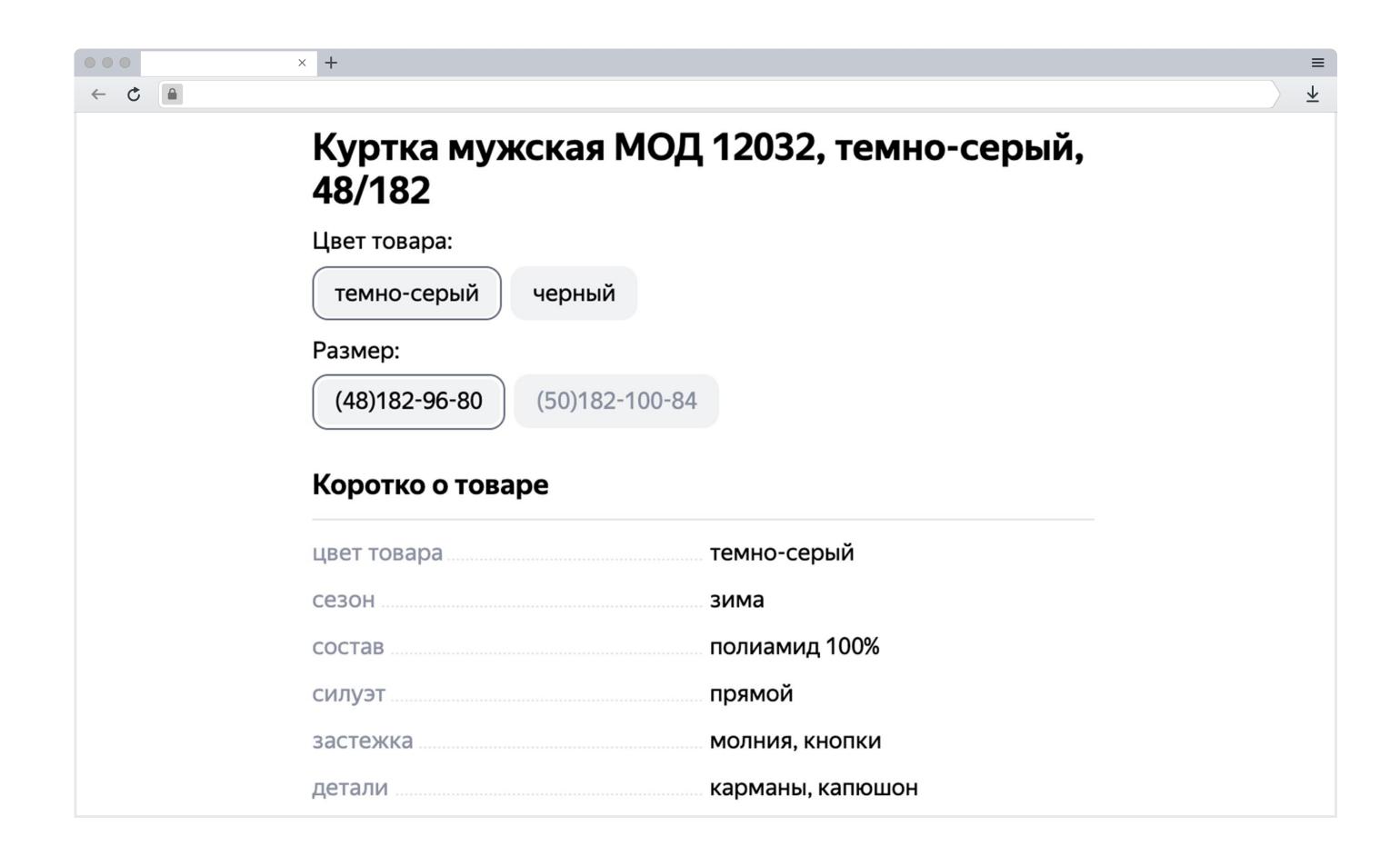


Что можно парсить ещё





Сложные характеристики товаров





Резюме

- + Несколько подходов к парсингу структурированных данных
- + Как поддерживать актуальность базы
- + Архитектура базы Товарного Поиска
- + Различные сферы применения алгоритмов парсинга



Резюме

- Н Несколько подходов к парсингу структурированных данных
- Н Как поддерживать актуальность базы
- + Архитектура базы Товарного Поиска
- + Различные сферы применения алгоритмов парсинга

Кучумов Илья

руководитель разработки Товарного Поиска @IlyaYndx

HL) HighLoad ** Яндекс

Обратная связь и комментарии по докладу по ссылке

